

Proposal for Google Summer of Code - 2017

KDE (Kopete)

Name : Vijay Krishnavanshi

Email : vijaykrishnavanshi@gmail.com

IRC : vijaykrishnavans

Mentor : Laurent Montel

Co - Mentor : Pali Rohar

Summary

1. Introduction
2. About the Project
3. Current State of the Project
4. Implementation
5. Timeline
6. Other Obligations
7. About Me

Introduction

Kopete is an Instant Messaging Client that is modular and has plugins. It requires KDE Plasma. Kopete has a huge codebase with lots of legacy code and reflects the hard work of many developers since many years. Kopete currently supports 9 different messaging services and has 15+ plugins.

About the Project :

Kopete is now ported to KF5. This project is about adding auto tests to Kopete. In this autotests for protocols and plugins will need to be written. This project also includes the improvement of support for protocols in KDE. This will be based on the discussions with the mentors and community. After this project a developer will be able to run all the tests for the complete project with help of the interface.

Current State of the Project :

I have already started working on this project. Some unit test cases which were earlier written for libkopete are not working. Focus will be able to get them working in few days. But the autotests for protocols and plugins need to be written from scratch. I have done testing integration in my SoK project also so that experience will be useful here also. I have made the interface in the SoK

projects also for testing which shows how it will be implemented for kopete. I am working to implement it in kopete now.

Implementation:

I have divided my project into several tasks to make the approach modular :

Task 0: This task will be to create the accounts and get the information together to make the Kopete testable as it has 9 protocols accounts in all these protocol will need to be set up.

Task 1: Work on certain remaining warnings and remove remaining certain compile warnings.

Almost the porting is complete of Kopete but still certain warnings are needed to be removed. This work will be done before the starting of project itself.

Task 2: Integrate make check command to the main project to create the interface for testing

This task will be get the make check command working for the project. This task will also include making the required changes to the some of the tests that are written for libkopete so that they also get executed when the make check command is run. At present, plan is that make check will take arguments to switch particular type of test cases on an off as required.

Task 3: Write autotests for the protocols

For writing tests QTest classes of Qt5 will be used. These tests will include unit testing of various components of protocols and tests of the protocols working with the main kopete application.

Task 4: Write autotests for the main application

For writing tests for main application also QTest will be used. And it will include unit testing of components of main application.

Task 5: Write autotests for plugins

Create a test server of kopete and test the plugins for their targeted work. There are around 15 plugins. Testing will cover unit tests and gui tests.

Task 6: Improve protocol support for kopete

This task will be based mostly on the discussion with the mentor and the community. There are todos listed in between the code and will implement them if possible.

Task 7: Create Hacking Manual for kopete to increase new developers contribution

Hacking Manual will help in getting the new developers to speed with the idea of kopete and how to understand its wide codebase. It will be useful for kopete in long run.

Documentation is spread out throughout the timeline. This will ensure the proper tracking of project at the time of review.

Deliverables:

Kopete Application with :

- Auto tests for Protocols
- Autotests for Plugins
- Autotests for Kopete main application
- Hacking Manual
- Improved support for protocols

Tentative Timeline

Community Bonding Period

This time will be utilised for:

- Work with community and discuss ideas for improving support for protocols
- Study extensively about the Qt Testing
- Trying out Qt Test in different settings and dummy projects
- Work on the remaining warning of Qt5 when compiling kf5 branch
- Understand changes need to be made in order to improve support of protocols.
- Write unit test cases for kopete main application.

I have exams in these first two weeks of may but I have adjusted timeline accordingly.

May 14 - June 26 (Phase-1 coding)

Community Bonding period will also be used for writing unit tests for main kopete and lib kopete. This work will continue throughout the project as needed.

May

- Week 3: This week will be utilised for writing unit tests for the main kopete application

- Kopete main application has many parts and for the testable components unit tests need to be written
- Some of it will be done in community bonding period.
- Week 4: This week will be utilised for writing unit tests cases for libkopete
 - Some unit tests are already written for this but more unit test cases will be written

June

- Week 1: Write autotests for the protocols
 - Write tests for the Protocols: Bonjour, Gadu and Groupwise
- Week 2: Write autotests for the protocols
 - Write tests for the Protocols: Jabber, Meanwhile and Oscar
- Week 3: Write autotests for the protocols
 - Write tests for the Protocols: Testbed, Wlm and Winpopup

Review Phase 1 :

- Week 4: Get it all together for evaluation
 - Documenting the testing process and changes made if any
 - Finish up the work and get the interface working for the testing of protocols, kopete main application and libkopete.

July 2 - July 29 (Phase-2 coding)

July

There are 15 plugins that need to be tested using unit testing or gui testing. It is divided into three parts 5 for each week. I think that this work will be complete before three weeks then the time will be give for the further improvement of the previously written test cases and discussion will be done on improving the protocol support.

- Week 1: Write auto tests for the plugins
 - Plugins: Latex, auto_replace, contact notes, text_effects and highlight
- Week 2: Write auto tests for the plugins
 - Plugins: highlight, alias, add_bookmark and privacy
- Week 3: Write auto tests for the plugins
 - Plugins: urlpicviewer, statistics, webpresence and otr
 -

Review Phase 2:

- Week 4: Get it all together for evaluation
 - Documenting the testing process and changes made if any

- Finish up the work and get the interface working for the complete application.
- By this time most of the testing part will be finished

August

- Week 1: Work on improving the protocol support in kopete application
 - There are certain todo in the source code will work on them
 - Do the work as discussed with mentor and community
 - Document the changes. Continue work on improving protocol support
- Week 2: Continue work on improving protocol support
 - This week will be for testing the application using real accounts on the protocols and find the errors
 - This week can act as buffer week to complete all the remaining task
- Week 3: Documentation and test new changes
 - Some of the debugging work can stretch to this week also.
 - Documenting the newly made changes
- Week 4: Work on Developer Hacking Manual
 - This week will focus on developing the hacking manual for the new developer in Kopete
 - Kopete has a wide codebase so it really needs one.

Other Obligations:

I have my semester exams in the first 2 Weeks of May (May 01 - May 14 tentatively), but I have adjusted my schedule accordingly. Also, I'll account for the delay if and when it arises by working extra during the coding period. I can commit more than 30 hours / week what is recommended for the project.

About Me:

I am 3rd year student of Computer Science and Engineering at Shri Mata Vaishno Devi University, Katra, India. I have been a KDE user for a couple of years and am contributing since last 6 months I took part in SoK and have worked on the KDE Now application during my project. I have successfully passed. My project was to write tests for KDE Now and integrate the calendar notification feature to it. During Seasons of KDE I learnt a lot about the KDE architecture of applications and Qt Tests.

I went through the code base of Kopete and am quite familiar with the work that need to be done for this project. I have started working on this project.

My Patches : <https://git.reviewboard.kde.org/users/vijaykrishnavanshi/?show-closed=1>